CMPG-767/CMPT-477 Image Processing and Analysis
Course Projects

The list of projects is given below.

Projects marked with * are for graduate students. Undergraduate students working on any of these projects may get **15** extra credit points. These projects can also be designed by mixed teams, which may include undergraduate and graduate students.

Projects marked with # are for undergraduate students only.

The following general requirements are common for all projects:

1) Students may work on projects in teams (up to 3 students in one team) or design projects on their own. Up to 5 students (graduate and undergraduate) may work in teams designing Projects 8 and 9

2) All projects should be resulted in a software system solving certain image processing problems

3) If a project is a team effort, all team members working on the project shall contribute to its design.

4) The following results are expected:

   a) **Each team (or individual)** will make a **10 min oral presentation** with slides showing a structure of a software system designed and with the demonstration of how it works. The **source code and slides are due the day of a final presentation**.
   b) **Every student** shall prepare **a written report** describing his/her contributions to the project and showing the results of test runs All **reports are due the day of a final presentation**.

5) **Grading criteria**:
   - Coding – **50 points** (for team projects – every single team member needs to make contributions in coding, otherwise he/she will not be able to get more than 20 points for this part of the project).
   - A report presenting own contributions and demonstrating how a system works – 3**0 points**
   - Final presentation – **20 points** (for each team member).

Project 1. Periodic Noise Filtering *

I. Design function utilizing Gaussian Median Notch filter for periodic noise filtering and a user interface (preferably GUI) for this function.

Functional requirements:

1) A function should be able to detect abnormal peaks in the power spectrum (magnitude of the Fourier transform) according to the median criterion and correct them according to the following rule $\tilde{C}(u,v)=\begin{cases} GaussNotch\{D(u,v)\} & \text{, if } \dfrac{D(u,v)}{MED\{D(u,v)\}}>T \\ D(u,v) & \text{, otherwise} \end{cases}$,

   where $D(u,v)$ is a magnitude of the Fourier coefficient of interest, and $\{D(u,v)\}$ is an $n\times n$ window around a magnitude of the spectral coefficient of interest $D(u,v)$.

2) A Gaussian Notch filter should be utilized through a component-wise multiplication $\{D(u,v)\}\circ\{G(u,v)\}$ of the $\{D(u,v)\}$ window of the Fourier transform magnitude by the following $n\times n$ Gaussian surface

$$\{G(u,v)\}=1-e^{-B\left[\left(u^2+\left\lfloor\frac{n-1}{2}\right\rfloor^2\right)+\left(v^2+\left\lfloor\frac{n-1}{2}\right\rfloor^2\right)\right]};\ u=-\left\lfloor\frac{n}{2}\right\rfloor,\dots,\left\lfloor\frac{n}{2}\right\rfloor; v=-\left\lfloor\frac{n}{2}\right\rfloor,\dots,\left\lfloor\frac{n}{2}\right\rfloor; B<1$$

3) A function should be able to process all coefficients of the Fourier transform magnitude, (including those located close to borders and on the borders) except the $\frac{n}{2}\times\frac{n}{2}$ area around a zero frequency Fourier coefficient $D(0,0)$, which should be left unprocessed.
4) A function should accept a noisy image to be processed, a window size **n**, and scaling coefficient **B** as calling arguments and return a filtered image as a processing result.

II. Test a designed function using several test images and measure PSNR wherever it is possible to evaluate the quality of filtering.

Project 2. Restoration of blurred images *

I. Design a function utilizing restoration of blurred images using a Wiener filter with regularization and a user interface (preferably GUI) for this function.

Functional requirements:

1) A function should be able to utilize a Wiener filter with regularization parameter α

$$\hat{F}(u,v)=\left[\circ\frac{\bar{H}(u,v)}{\left|H(u,v)\right|^2+\alpha K}\right]\circ G(u,v),$$

where $H(u,v)$ is a Fourier transform of PSF, $G(u,v)$ is a Fourier transform of an image to be restored, $K\approx\frac{\sigma^2_{Noise}}{\sigma^2_{image}}$ or $\approx\frac{(NM)\sigma^2_{Noise}}{\sigma^2_{image}}$ (depending on normalized or non-normalized Fourier transform is used) is a noise-to-signal power ratio, N x M is an image size, $A\circ B$ is a component-wise matrix multiplication of matrices A and B and $\circ\frac{A}{B}$ is a component-wise division of matrices A and B.

To estimate σ^2_{Noise}, a blurred image should be filtered using, for example the rank order EV filter or smart linear filter and then a resulting image should be subtracted from the blurred one. The difference should be used as an approximation of noise and its standard deviation σ_{Noise} and variance σ^2_{Noise} should be found.

2) * To draft a solution, a Wiener filter can be utilized using the **deconvwnr** function from Matlab.
* A PSF should be generated using the **fspecial** function from Matlab.
* Blur can be simulated using the **imfilter** function from Matlab.

3) A function should accept a blurred image to be processed, and a regularization parameter α, and return a filtered image as a processing result.

II. Test a designed function using several test images, try to find a value of α leading to the best result and measure BSNR and ISNR wherever it is possible to evaluate the quality of a restored image.

Project 3. A software system for linear and nonlinear image filtering in the spatial domain *

I. Design a software system with a graphical user interface (GUI) performing linear and nonlinear filtering in the spatial domain.
Working with color images, this system should be able to process either only a luminosity channel or each of RGB channels separately.

II. Your software system should include a function for evaluation of statistical characteristics of images (mean, standard deviation, RMSE and PSNR) and a function for noise modeling (additive Gaussian noise and random impulse noise)

III. The following filters should be implemented:

1) A linear convolution with a 3x3 window (a user should be able to specify a kernel by loading it from a text file or specifying kernel weights directly)
2) Mean filter with a 3x3 window
3) Smart filter with a 3x3 window
4) Median filter with a 3x3 window
5) Impulse noise filtering with the differential rank impulse detector (DRID) followed by median filtering of pixels where noise has been detected
6) Rank-order EV filter (mean) with a 3x3 window
7) Rank-order ER filter (median) with a 3x3 window
8) Unsharp masking with a 3x3 window
9) Sobel, Laplace I and Laplace 2 edge detectors

IV. Test a designed system

Project 4. A software system for linear image filtering in the spatial domain

I. Design a software system with a graphical user interface (GUI) performing linear filtering in the spatial domain.
Working with color images, this system should be able to process either only a luminosity channel or each of RDB channels separately.

II. Your software system should include a function for evaluation of statistical characteristics of images (mean, standard deviation, RMSE and PSNR) and a function for noise modeling (additive Gaussian noise and random impulse noise)

III. The following filters should be implemented:

1) Mean filter with a 3x3 window
2) Smart filter with a 3x3 window
3) Unsharp masking with a 3x3 window
4) Sobel and Laplace 2 edge detectors

IV. Test a designed system

Project 5. A software system for nonlinear image filtering in the spatial domain #

I. Design a software system with a graphical user interface (GUI) performing nonlinear filtering in the spatial domain.
Working with color images, this system should be able to process either only a luminosity channel or each of RDB channels separately.

II. Your software system should include a function for evaluation of statistical characteristics of images (mean, standard deviation, RMSE and PSNR) and a function for noise modeling (additive Gaussian noise and random impulse noise)

III. The following filters should be implemented:

1) Median filter with a 3x3 window
2) Impulse noise filtering with the differential rank impulse detector (DRID) followed by median filtering of pixels where noise has been detected
3) Rank-order EV filter (mean) with a 3x3 window

II. Test a designed system

Project 6. A software system for edge detection and sharpening

I. Design a software system with a graphical user interface (GUI) performing edge detection and sharpening using the following operators:
Working with color images, this system should be able to process either only a luminosity channel or each of RDB channels separately.

1) Laplace I for downward intensity jumps
2) Laplace I for upward intensity jumps
3) Directional Laplacian edge detectors
4) Sobel (horizontal and vertical intensity jumps)
5) Unsharp mask

II. Test a designed system

Project 7 Research Project

Discover Fourier power spectra of the following linear filters' 3x3 kernels:

1) Mean filter
2) Smart filter
3) Unsharp mask (with k =2)
4) Laplace edge detector upward 1
5) Laplace edge detector downward 1
6) Laplace edge detector upward 2
7) Laplace edge detector downward 2
8) Directional Laplacian edge detectors
9) Sobel (horizontal and vertical intensity jumps)

Make your conclusions about the frequency areas, which each filter passes, suppresses, and possibly enhances confirming these conclusions by respective illustrations.

Project 8 (up to 5 students may work), 15 extra credit points *

Project 1 + Project 2 in a single software system

Project 9 (up to 5 students may work), 25 extra credit points *

Project 1 + Project 3 in a single software system

Or

Project 2 + Project 3 in a single software system